ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil

Modeling an Active Fault-tolerant Control for Manufacturing System.

Joao Thiago de Guimaraes Anchieta e Araujo Campos
Carlos Cezar Rodrigues Santos
Carlos Heitor Filgueiras de Souza
Robson Marinho da Silva
Santa Cruz State University
joaothiagocampos@gmail.com, carlosrodrigues.eps@gmail.com, carlosheitor04@gmail.com, rmsilva@uesc.br

Abstract. *A flexible manufacturing system (FMS) is an integrated system composed with work stations controlled by computers which permit the use of a same machine to perform different processes. This flexibility increases the level of occurrence of faults due to the several changes of process sequences. Hence, the control of these processes must be highly precise to avoid the failures possibilities and, consequently, decreasing problems on a production line such as machine damages, defective products, etc. Generally, the failure is detected by a human operator which may analyse the failure erroneously and do not finding the right solution. Then, it is important a good control system to support the human decision on a detection, control and correction of a FMS. To improve the flexible processes, the use of a modelling method before implementing the FMS, through simulation of different scenarios, is necessary because it reduces the possibility of damage on machines, testing and analyzing the possible faults which may occurs. Among several modeling methods of FMS, the Petri net (PN) have an easy graphic visualization where is possible to model mechanisms for fault detection, control and decision-making and to facilitate the conversion to language of the most programmable logic controllers (PLC). Thus, this research use the PN modelling technique to create a model of active fault-tolerant control system applied on a FMS didactic station using Petri net models converted to Grafcet language, showing the advantages of the proposal to improve the dependability of FMS.*

Keywords: *petri net, manufacturing system, fault tolerant, production flow schema, grafcet... (up to 5 keywords)*

1. INTRODUCTION

In current economic paradigm, the optimization of production systems are no longer seeking to reduce time of production but to reduce machine maintenance operations (MMO). Additionally, there is several problems which may interrupt production process, such as faults in production line, machine breakdowns, unscheduled downtimes or humans mistakes. Indeed, when a MMO occurs, it generates a production idle time stopping production line, generating high costs to factory. In a production system, identifying causes which may stop production is a hard task. It is a complex decision to identify all variables which may influence on the production problems such as machines, information technology or human behaviour. To grant reliability of production processes, it is necessary to execute preventive and corrective maintenance operations based on machines breakdowns. Then, MMO maintain high reliability of equipments and avoid production idle time (Wu and Ni (2011); Boschian *et al.* (2008); Simeus-Abazi *et al.* (2010)). Even a production planning consider MMO, it is not possible to predict every possible faults, then a good prevent maintenance planning improve process production instead of executing any corrective actions which causes productions problems (Schouten and Vaneste, 2010; Boschian *et al.*, 2008). Hence, a manufacturing system must consider an integrated planning of production and maintenance to optimize production in a trade-off relation (Berrichi *et al.* (2010); Nourelfath and Chatelet (2012); Chung *et al.* (2009)).

In automated manufacturing such as Flexible Manufacturing System (FMS), MMO have a more important role in costs optimization because the operator decisions over an automatic production system are less effective than on traditional manufacturing which has a high dependency of operator decisions. Generally, a human operator detects fault based on the exact moment of failure which is a mistake because faults occur previously and failures may be a result of a fault chain reaction ((Chiodo *et al.*, 2004)). Hu *et al.* (1999) says that generally 80 % of downtime caused by a fault is spent locating its source and about 20 % is spent on repair. Thus, even faults may be from physical or human origin, on a high automatized manufacturing level it is more dependent of machines and computers than human decisions. Faults detected by human are subjective then, if possible, it must be also controlled by a control system to avoid a MMO and, consequently, reducing maintenance costs. A misunderstanding of informations provided by the human interface device (HID) generates an incorrect situation assessment by the operator and an erroneous knowledge will not detect fault in time and, consequently, a severe failure in production system (Bley, 2002; Chiodo *et al.*, 2004; Patwardhan *et al.*, 2006). Hence, faults caused by automatized machines operations added to operators misunderstanding production informations must be avoided in all production system, specially on FMS. A solution for failures situations is to share decision support between operators and automatic decision support systems. Those systems will help to identify faults correctly and take an automatic corrective action, if it occurs, without depending of human analysis and, further, permit the use of human experience and intuition to analyse a consequence of these decisions if it do not collaborate to optimization production system or any information system that is not inserted on the system decision support (Mendonca *et al.*, 2012).

Considering faults on a decision support system help to reduce the use of MMO and improve optimization process. The control system must be a detection and diagnosis tool. By detecting failure, a diagnosis system must be used to identify correctly the faults sources and correct it automatically by avoiding process shut-downs from simple faults (Yu and Yu, 2005; Berrios *et al.*, 2011). Even a corrective maintenance operation generates unnecessary costs to factory because it is fixing a manufacturing problem undetected by the decision support or operators, the use of a preventive maintenance plan will avoid those errors but it also generates preventive costs which is possible to eliminate by a good decision support tool. Thus, the faults must be considered on all preventive decisions. On the FMS context, where it is highly automated, it is possible to develop an active fault tolerant control to minimize costs of MMO uses.

Blank *et al.* (2001) defined the fault tolerance as the ability of a controlled system to maintain control objectives, despite the occurrence a fault. Thus, The fault tolerant control has an important role on any FMS. Even human decisions is based on intuition and experience, the system must integrate the operator and fault tolerant control as a efficient decision support. The integration of information provided by processes and human knowledge results in a good fault tolerant system (McNaught and Chang, 2011). By using an automatic control, if faults detection and diagnosis procedures are not well defined previously, the decisions by a fault tolerant control may cause several damages on machines, human health or environment (Blank *et al.*, 2001). To grant the system control working as requested, it must be reliable and efficient. According to Adamyan and He (2002), the system reliability does not depends only of fail states but also on the sequence of occurrences of failures. Hence, it must be seen has a whole production system which must consider the production factors, transformation processes and production requirements. A control system grant the reliability and system safety by using intelligent decisions support considering the fault occurrences. Then, the automatic control must be safety with no errors before it is implemented into a production system. Furthermore, a fault tolerant control must be flexible enough to be easily changed while production without any interruptions, then a reconfigurable behaviour will also be considered.

Therefore, developing a fault tolerant control system as a tool for decision support is complex to any manufacturing system. If it is implemented without any validation method or safety system, which prevent production system damages or grant system reliability, a complete system degradation will generates failures, faults and production problem that are constraints to optimization process. Then, because of this complexity level, the use of methods and techniques to validate those systems must be used before the implementation to manufacturing. To ensure the production operations, a model must be developed and, if it works as required, it will be inserted into the whole system which integrates information, manufacturing and productions systems(Patwardhan *et al.*, 2006; Hu *et al.*, 1999; Vilanni *et al.*, 2005).Vineyard *et al.* (1999) confirm that a model help to identify the real complexity of system failures and understanding how these systems operate in practice. Further, a model permit to improve the control and every systems integrated to it such as production and information systems which are directly linked to the control specifications.

A fault tolerant control represents the possible situations which a fault appears considering solutions to keep production system working. In cause of failure, the model must identify the fault source, diagnoses the resulting error and execute a solution to solve the problem with a minimum of production idle time. On optimization context, it must execute a correction quickly enough to not stop production and keep the process working normally. This work may be executed or by operator, or by computers decisions. In case of a fault tolerant control, the decision must eliminate human behaviour problems for faults diagnosis decision. This control is inserted into computer aided manufacturing (CAM) and executed over a programmable logic controllers. Using a set of computers to control a decision support tool is more reliable than operator decisions, then, by a good fault tolerant model, it is possible to eliminate failures caused by undetected faults.

The model technique used to improve developing fault tolerant control system have to represent correctly the dynamic of detection and diagnosis system, and easily converted to programmable logic controllers (PLC). PLC are almost used on manufacturing system, specially on FMS because of its adaptive, modular and user-friendly properties (Hu *et al.*, 1999). Even this decision support is executed by computers before sending informations to machines/tools, the PLC is the communication between physical equipments/machines and the fault tolerant model. After developing the information system decision model, it must integrate and communicate to computers which will be used as a computers decision center and to PLC where it will execute decisions results. Human behaviour are also considered on this model by supervising all decisions made by the automatic control system. A good model technique must be selected because it will communicate among those decisions factors and grant a safety, reliable and quickly information sharing communication. For example, the information provided by production line will be treated according to model information flow, defined previously by the model technique and method applied.

The Grafcet language is used on this research because of the huge application on industry. Specially on discrete event systems (DES), this language has many advantages over others traditional language such as graphical visualization and its uses with LADDER language which is almost used by PLC manufacturers (Moura and Guedes, 2012; Yan and Zhang, 2010; Weltch, 1992). Then, the model developed must be easily converted to Grafcet and Ladder diagrams, which will control the production line of a manufacturing system. Further, the selected modeling technique needs to be implemented integrating human behaviour by providing informations to HDI. Hence, the figure 1 illustrates the model and implementation relations of a manufacturing system.

ISSN 2176-5480

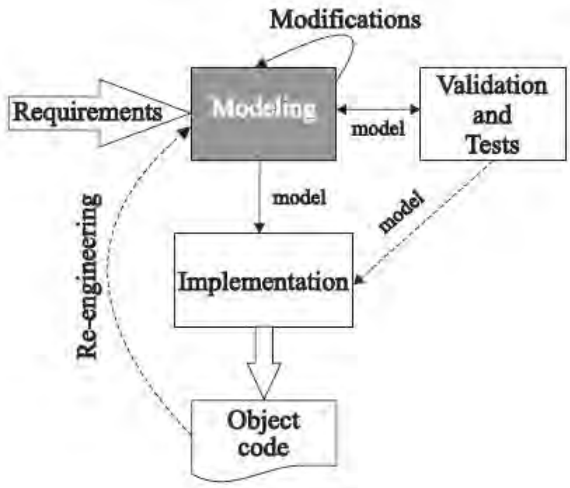

Figure 1. Modelling Implementation overview. Source: Moura (2012).

The Petri nets (PN) are a good modelling technique to develop a fault tolerant control. Because of its easy graphical visualization and based on linear algebra, which shows its power to model complex systems, PN may design an effective and optimized control model to any kind of manufacturing systems (Silva *et al.*, 2011; Murata, 1989a). This technique is normally used on software developing models but it is highly recommended to systems in general. By using PN, it is possible to analyse a system behaviour and develop a model to fault detection, control and correction. PN also facilitates the implementation, because its similarities to IEC 60848 GRAFCET language, a wide used PLC language. However, a method to correctly use the PN to manufacturing system must be defined. Even PN is an effective technique to manufacturing system, it is important to consider others variables which is not explicit on PN models, then, associating PN with Production Flow Schema (PFS) extension is a good modelling method because it is possible to identify all kind of variables, quantitative or qualitative, and further easily implemented on PN final model. Hence, this study developed a method to model a fault tolerant control using PN and PFS applied on a high automated manufacturing system which may be considered as the new paradigm of Flexible Manufacturing System (FMS) and converted to Grafcet PLC language.

2. FUNDAMENTAL CONCEPTS

2.1 Manufacturing System

A manufacturing system involves many aspects as labor-intensive production systems, mass-production line, automated mass-production line, job shop and group/cellular manufacturing cells to flexible manufacturing system. Historically, this manufacturing changes follow the paradigms evolution by a gradual migration from "manufactum" production to reconfigurable systems. The main change from these scientific management behaviour is the use of information systems to improve productions systems. While the mass production paradigm search for reducing costs of production, researchers have elaborated applied studies to improve production system by considering others variables such as human behaviour, work environment, factory layout, etc. Buzacott (1995) list the manufacturing evolution from old paradigm to new paradigm. The old paradigm began with works of the scientific managements movement, such as F.W. Taylor and Gilbreth. This production concepts was divided on follows attitudes and ideas: i) assigned tasks to each operator of factory, ii) Staff provides ideas and solve problems, iii) Managers are in charge to ensure that tasks got done, iv) evaluation based on quantity and direct costs and v) centralized coordination. Meanwhile, the new manufacturing systems have they evolutions based on technologies development. The arrival of computers has initiated a constant growth of flexible paradigm. The programmable logic controllers (PLC) permit to increase the production factors flexibility by using the same tools and machines to execute different processes. Then, it is possible to associate mass production, quality and production variety on a new paradigm denominated reconfigurable manufacturing system. Thus, the flexible manufacturing system is responsible for production diversity. By responding customers needs, the FMS is an innovative way which integrated information system, flexible processes and computers control.

The high level of automation increases the customizable production as high productivity market needs. However, FMS developed in the last two decades: (i) are expensive, since in many cases they includes more functions than needed, (ii) utilize inadequate system software, since developing user-specified software is extremely expensive, (iii) are not high reliable and (iv) are subjective to obsolescence due to advances in technology. To overcome this limitation a reconfigurable manufacturing paradigm has a more adaptable perspective, in which it is adjustable to the business and market interest. The reconfigurable manufacturing system (RMS) must ensure the functionalities considering the use of mechatronic and

ISSN 2176-5480

computer-aided technologies and, thus, adjustable to market interest.

2.2 Programmable Logic Controller (PLC)

A PLC is an hardware interface created to control machines and/or processes executing programs hosted in their memory and using a simple computer language. It is divided in two parts: a processor which host the PLC program into memory and the I/O interface (I - entry of signal(s), O - Exit of signal(s) to receive or send signals respectively). By using a binary language, the signals are converted to a logic computer language which responses with new signals(s) addresses to the O ports. On RMCS, Ladder diagram and GrafCet (Figures 2 and 3) are too commons language wide used. While Ladder diagram has a simple electric circuits notation, GrafCet uses of state diagram to set up a list of conditions to execute a new transition and thus creating a new condition state, closely to Petri net representation.

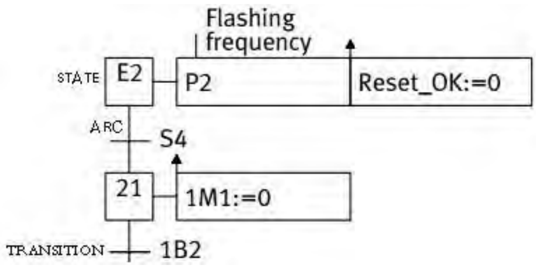

Figure 2. Modelling Implementation overview. Source: Moura (2012).

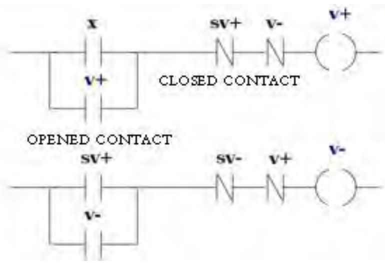

Figure 3. Ladder diagram example

On the GrafCet diagram example in figure 2, the state determines which value the Output signals must assume. To change an O signal, the state should have an arc connecting it to a transition. The arc give the direction of the new transition which determines the conditions to keep reading the language. The LADDER diagram uses of eletric symbols to execute the PLC program. The symbol below the label v+ indicates an opened contact and, when activated, will assumes a closed contact. Similarly to the label v+, the symbol below v- indicates a closed contact that will be opened if it is activated. The GrafCet has an easy graphic visualization and it is more intuitive than LADDER diagram. Both languages are widely used but for this study it will be used GrafCet as the PLC language.

2.3 Petri net (PN) and its extensions

The Petri net (PN) concept was created by Carl Adam Petri has an easy mathematical and graphical modelling technique. Initially was used to describe the relation of conditions and events on protocols studies (Silva (2008)). However, this modelling technique has been expanded to a wide research fields on manufacturing system, specifically on automation and discrete event system (EDS).
PN are bipartites graphs compound of places (represented by circles), transitions (represented by bars), oriented arcs (arrows) and marks (black dot). Arcs are oriented either from a place to transition or from a transition to a place, in a graphical representation (Murata (1989b), Silva (2008)). Considering the arc directions, input place of a transition is the place which have an arc oriented to it and an output place is where an oriented arc begin to be oriented to another place. The figure 10below shows examples of places, transitions and arcs:

By mathematical representation, a PN is a four-tuple represented by a set of places, transitions, arcs and weight. If considering the initial mark of a PN, it may be inserted into the four-tuple as a new variable of a PN which become a five-tuple, PN = (P, T, F, W, M_0) where :
P = p_1, p_2,..., p_n is a finite set of places,
T = { t_1, t_2,..., t_n} is a finite set of transitions,
F $\subseteq$ (P x T) $\cup$ (T x P) is a set of arcs (flow relation),

ISSN 2176-5480

W:F $\rightarrow$ {1,2,3,...} is a weight function,
M_0:P $\rightarrow$ {0,1,2,3,...} is the initial mark,
P $\cap$ T = ϕ and P $\cup$ U $\neq$ ϕ.
A Petri net structure is denoted by N = {P,T,F,W} without any initial marking.

This modelling technique may be used to develop both abstract or physical models . For example, a manufacturing model should have different levels of abstraction. While it is possible to describe the machine-tools equipment and their relation, by using PN it is possible to modelling dynamic system which integrates the process to information flow, decision support, etc. The basic rules of PN modelling are:

- A transition may only be fired if the input places connected to this transition are marked.
- The firing transition removes an equal number of the arc weight from input places and add it to the output place.
- Marks do not have any identification, it is used only to determine if the condition is true or false.

Petri nets represents any process dynamic as far as graphical visualization or mathematical model. It permits to: (i) represent different kind of systems, (ii) mathematical descriptions of the model, (iii) easy graphical representation, (iv) parallelism, synchronization and other dynamic processes behaviour, (v) representation of static or dynamic elements; and (vi) use of methods and analysis tools and simulation. The Figure 4 shows some typical processes represented by PN.

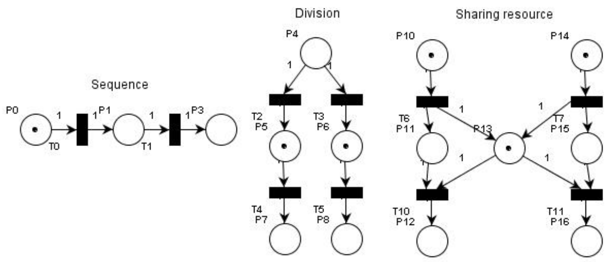

Figure 4. Petri net examples.

3. Method

The method used in this reasearch is divided in three steps: Step one - problem definition by a functional fluxogram; Step two - modelling using Petri net, and Step three - conversion of PN model to GrafCet language.

Step one - problem definition by a functional flowchart: the aim of this step is to identify and integrate the detection and control tools to the modelling requisites as sensors, actuators and the dynamism of the study process. It must be defined all equipments of the production processes evolved and the inputs/outputs flow. By integrating this information, it is possible to understand the production system which will be used on control system model.The flowchart technique allow to identify the possible faults situations and which equipments/tools are evolved on detection (sensors) and correction (actuators) processes.

Step two - modelling using Petri net: Due to model complexity, it is necessary to guarantee the functionality of the model process before applying into graphic model visualization. Thus, the model technique is divided in: (i) creation of a conditions/actions (c/a) list of each process step determined on flowchart, (ii) conversion of c/a list to matrices representations which must be divided on input and output matrices, (iii) graphic visualization of PN using the matrices representations.

Step three - further, based on the PN model, the GrafCet conversion will be done following the flowchart sequence and the PN model conditions. The flowchart shows all input and output variables for each process step and pn models indicate how it must be developed to execute the process and control as required by the system control.

4. Discussion and results

The study object is a handling piece production process (Figure 5) divided on two steps. Firstly, a cylinder push the piece from a buffer to a magazine location which it waits for the next step. If there is no piece on buffer, an operator must insert new pieces manually. Secondly, the piece is moved from magazine to a next production stage by a changer module, also automatic. On the second process there is no operators intervention which means that all processes are executed automatically by the module changer. The macro-process in study is represented on Figure 6. Thus, by following the method steps, it was created a functional flowchart to determine the process flow and all equipments involved on each

ISSN 2176-5480

step. Based on flowchart, it is possible to identify the sensors, controllers and actuators which control each sub-process of the handling piece process. Even each step is a part of the whole production system, this step is very important to identify when the control system must be activated to diagnostic the error and, if necessary, correct it automatically without an operator intervention.

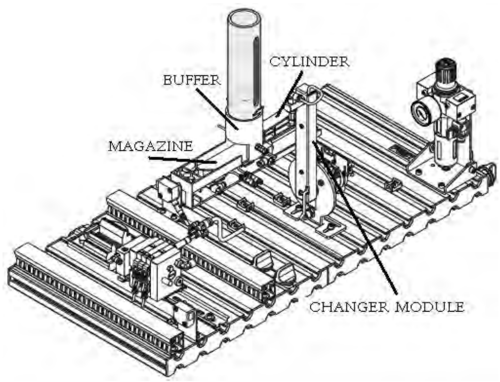

Figure 5. Handling piece production process

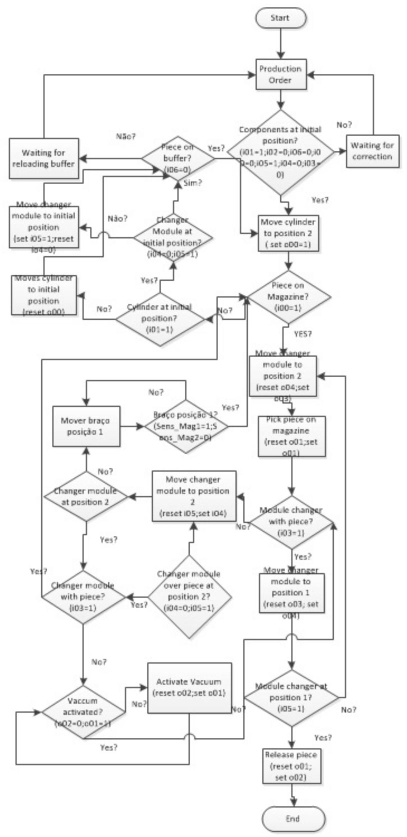

Figure 6. The macro-process flowchart

Each flowchart state is followed by a question where determine the sensors state to grant the process working. The sensors are identified following the PLC I/O ports names where it is connected to. If there is any fault while the process, the flowchart indicates it by the "no" path and then start a correction or verification sub-process. Hence, this step is very important to identify which sensors (identified by input PLC ports) are evolved and, consequently, which actuators (identified by Output PLC ports) must be executed to continue the process normally.

Based on states and actions of the previously step, it is possible to develop the PN model. According to step two of the method, the Table 1 represents the conditions and actions of macro-process without any detection, control and correction control system. Then, it is converted to a matrix representation where is possible to identify the information flow or process flow in Figure 7. Further, it is possible to model the control system identified on flowchart (step 1). Initially,

ISSN 2176-5480

the cylinder control system is created a list (Table 2) representing it on matrix in Figure 8. Finally, the module changer must be modelled also. Below are the conditions/actions Table 3 and the mathematical model represented with two matrices (Figure 9), input and output respectively. By representing each input and output matrices with the PN graphical visualization, it results on the PN model as shown on Figure 10.

Table 1. Conditions and actions of macro-process

Conditions		Actions	
P0	Initial Position	T0	Cylinder push piece to magazine
P1	Magazine with piece	T1	Move CM over piece
P2	CM over magazine with piece	T2	Move CM to next station
P3	CM over next station	T3	Release piece
P4	Piece released	T4	Restart process

Figure 7. Input and Output matrices of Macro-process

$$
\begin{bmatrix}
 & T0 & T1 & T2 & T3 & T4 \\
P0 & 1 & 0 & 0 & 0 & 0 \\
P1 & 0 & 1 & 0 & 0 & 0 \\
P2 & 0 & 0 & 1 & 0 & 0 \\
P3 & 0 & 0 & 0 & 1 & 0 \\
P4 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
 & T0 & T1 & T2 & T3 & T4 \\
P0 & 0 & 0 & 0 & 0 & 1 \\
P1 & 1 & 0 & 0 & 0 & 0 \\
P2 & 0 & 1 & 0 & 0 & 0 \\
P3 & 0 & 0 & 1 & 0 & 0 \\
P4 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
$$

Table 2. Control system list

Conditions		Actions	
P0	Initial Position	T0	Cylinder push piece to magazine
P5	Cylinder control system activated	T5	Activate cylinder correction system
P6	Cylinder Initial position sensor	T6	Request sensors state of cylinder correction system
P7	Cylinder Initial position sensor fixed	T7	Execute cylinder actuator to initial position
P8	Cylinder final position sensor	T8	Verify cylinder initial position sensor
P9	Cylinder final position sensor fixed	T9	Execute cylinder actuator to initial position
P10	Piece on buffer	T10	Verify cylinder final position sensor
P11	Empty magazine		
P12	Correction system starting state P6		
P13	Correction system starting state P7		

Figure 8. Input and Output matrices os control system.

$$
\begin{bmatrix}
 & T0 & T5 & T6 & T7 & T8 & T9 & T10 \\
P0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
P5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
P6 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
P7 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
P8 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
P9 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
P10 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
P11 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
P12 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
P13 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
 & T0 & T5 & T6 & T7 & T8 & T9 & T10 \\
P0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P5 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
P6 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
P7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
P8 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
P9 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
P10 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
P11 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
P12 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
P13 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\tag{1}
$$

Figure 9. Input and Output of module changer

$$
\begin{bmatrix}
 & T1 & T11 & T12 & T13 & T14 & T15 & T16 & T17 & T18 \\
P1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P14 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
P15 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
P16 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
P17 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
P18 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
P19 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
P20 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
P21 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
P22 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
P23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
 & T1 & T11 & T12 & T13 & T14 & T15 & T16 & T17 & T18 \\
P1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P14 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
P15 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
P16 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
P17 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
P18 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
P19 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
P20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
P21 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
P22 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
P23 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\tag{2}
$$

ISSN 2176-5480

Table 3. Condition and actions of module changer

Conditions		Actions	
P1	Piece on magazine	T1	Move module changer over magazine
P2	CM over magazine with piece	T11	Activate CM correction system
P14	CM correction system activated	T12	Request sensors state of CM correction system
P15	CM Initial position sensor	T13	Execute CM actuator to initial position
P16	CM initial position sensor fixed	T14	Verify CM initial position sensor
P17	CM next station sensor	T15	Execute CM actuator to initial position
P18	CM next station sensor fixed	T16	Verify CM final position sensor
P19	Vacuum sensor	T17	Activate vacuum
P20	Vaccum activated	T18	Verify vacuum sensor
P21	Correction system starting state P15		
P22	Correction system starting state P17		
P23	Correction system starting state P19		

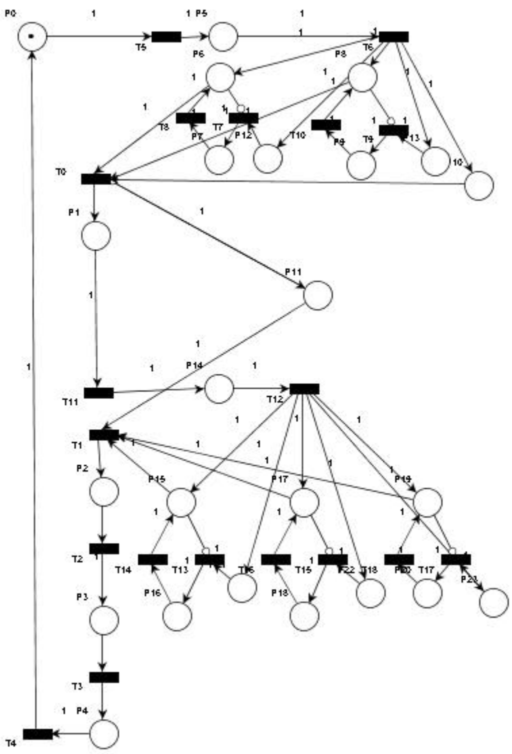

Figure 10. Petri net of the control system

ISSN 2176-5480

Finally, the PN is converted to GrafCet language. It is possible to converted each state/action situation shown on PN to GrafCet (Figure 11) .

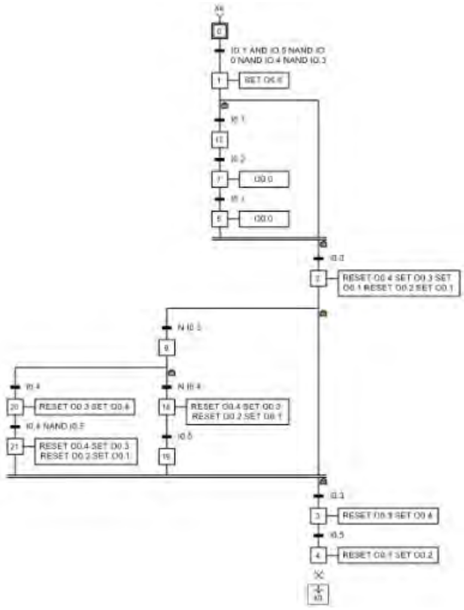

Figure 11. GrafCet of the control system

The state and action of GrafCet is similar to PN but the logic structure is different. For example, the control system of cylinder or module exchanger must be inserted into GrafCet as a series circuits conditions even it seems to be parallel on PN model.

5. Conclusion

The study presents a method developed which is based on Petri nets (PN) technique applying on a handling products process of a reconfigurable manufacturing system (RMS). The RMS use the same tools to do different processes and thus increase the flexible production. To control this automated system it is important to model it before implemented to industry because it reduce costs and avoid problems to production lines. The method demonstrates how each input and output variables that must be controlled on RMS to detect and verified in a production process can be used and applied as using a tool for decision support in a fault-tolerant control system. The use of Petri net is the best choice to model a control system by its easiness to graphical visualization and similarity to GrafCet language, an IEC60848 language. Tools for editing PN can be to modeling, simulation and validation of models. The developed PN models are converted to GrafCet to program programable logic controller (PLC) of the handling products system and thus the study demonstrates how to facilitate the implementation phase in industry. The study herein concentrates on modeling including the development to requirements specifications to machine operation and maintenance, to assure flexibility, efficiency and robustness. The method is generic and developed based on RMS, it can be tailored for specific another applications.

6. REFERENCES

Adamyan, A. and He, D., 2002. “Analysis of sequential failures for assessment of reliability and safety of manufacturing systems”. *Reliability Engineering and System Safety*, Vol. 76, pp. 227–236.

Berrichi, A., Yalaoui, F., Amodeo, L. and Mezghiche, M., 2010. “Bi-objective ant colony optimization production and maintenance scheduling”. *Computers and Operational Research*, Vol. 37, pp. 1584–1596.

Berrios, R., Nunez, F. and Cipriano, A., 2011. “Fault tolerant measurement system based on takagi-sugeno fuzzy models for a gas turbine in a combined cycle power plant”. *Fuzzy Sets and Systems*, Vol. 174, pp. 114–130.

Blank, M., Staroswiecki, M. and Wu, N., 2001. “Concepts and methods in fault-tolerant control”. In *Proceedings of the American Control Conference*. Arlington, United States of America.

Bley, D., 2002. “New methods for human reliability analysis”. *Environmental management and Health*, Vol. 3, nÂž 3, pp. 277–289.

Boschian, V., Rezg, A. and Chelbi, A., 2008. “Contribution of simulation to the optimization of maintenance strategies for a randomly failing production system”. *European Journal of Operational Research*, Vol. 197, pp. 1142–1149.

ISSN 2176-5480

Buzacott, J.A., 1995. "A perspective on new paradigms in manufacturing". *Journal of Manufacturing Systems*, Vol. 14, no 2.

Chiodo, E., Gagliardi, F. and Pagano, M., 2004. "Human reliability analyses by random hazard rate approach". *COMPEL: The international Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol. 23, nÂž 1, pp. 65–78.

Chung, S.H., Law, H.C.W. and Ho, G.T.S., 2009. "Optimization of system reliability in multi-factory networks by maintenance approach". *Expert Systems with Applications*, Vol. 36, pp. 10188–10196.

Hu, W., Starr, A.G. and Leung, A.Y.T., 1999. "Two diagnostic models for plc controlled flexible manufacturing systems". *Machine Tools and Manufacture*, Vol. 39, pp. 1979–1991.

McNaught, K. and Chang, A., 2011. "Bayesian networks in manufacturing". *Journal of Manufacturing Technology Management*, Vol. 22, nÂž 6, pp. 734–747.

Mendonca, L.F., Souza, J.M.C. and da Costa, J.M.G.S., 2012. "Fault tolerant control using a fuzzy predictive approach". *Expert Systems with Applications*, Vol. 39, pp. 10630–10638.

Moura, R.S. and Guedes, L.A., 2012. "Using basic statechart to program industrial controllers". *Computer Standards and Interfaces*, Vol. 34, pp. 60–67.

Murata, T., 1989a. "Petri nets: Properties, analysis and applications". *Proceedings of IEEE*, Vol. 77, no 4.

Murata, T., 1989b. "Petri nets: properties, analysis and applications". *Proceedings of the IEEE*, pp. 541–580.

Nourelfath, M. and Chatelet, E., 2012. "Integrating production, inventory and maintenance planning for a parallel system with dependent components". *Reliability Engineering and System Safety*, Vol. 101, pp. 59–66.

Patwardhan, S.C., Manuja, S. and Shah, S.L., 2006. "From data to diagnosis and control using generalized orthonormal basis filters. part ii: Model predictive and fault tolerant control." *Journal of Process Control*, Vol. 16, pp. 157–175.

Schouten, F.A.V.D. and Vaneste, S.G., 2010. "Maintenance optimization of a production system with buffer capacity". *European Journal of Operational Research*, Vol. 82, pp. 323–338.

Silva, R.M., 2008. "Modeling of intelligent building control systems considering fault occurrences". *Master Dissertation, University of Sao Paulo, Sao Paulo (in portuguese).*

Silva, R.M., Junqueira, F., Filho, D.J.S. and Myiagi, P.E., 2011. "Design of reconfigurable and collaborative control system for productive systems". In *Proceedings of COBEM 2011*. Natal, Brazil.

Simeus-Abazi, Z., Mascolo, M.D. and Knotek, M., 2010. "Fault diagnosis for discrete event systems: Modelling and verification". *Reliability Engineering and System Safety*, Vol. 95, pp. 369–378.

Vilanni, E., Pascal, J.C., Myiagi, P.E. and Valette, R., 2005. "A petri net-based object-oriented approach for the modelling of hybrid productive systems". *Nonlinear Analysis*, Vol. 62, pp. 1394–1418.

Vineyard, M., Amoako-Gyampah, K. and Meredith, J.R., 1999. "Failure rate distributions for flexible manufacturing systems: An empirical study". *European Journal of Operational Research*, Vol. 116, pp. 139–155.

Weltch, J.T., 1992. "Translating unrescricted relay ladder logic into bollean form". *Computers in Industry*, Vol. 20, pp. 45–61.

Wu, Q. and Ni, Z., 2011. "Car assembly line fault diagnosis based on triangular fuzzy support vector classifier machine and particle e swarm optimization". *SExpert systems with applications*, Vol. 38, pp. 4727–4733.

Yan, Y. and Zhang, H., 2010. "Compiling ladder diagram into instruction list to comply with iec 61131-3". *Computers in Industry*, Vol. 61, pp. 448–462.

Yu, D.L. and Yu, D.W., 2005. "Adaptative neural model-based fault tolerant control for multi-variable processes". *Engineering Applications of Artificial Intelligence*, Vol. 18, pp. 393–411.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.